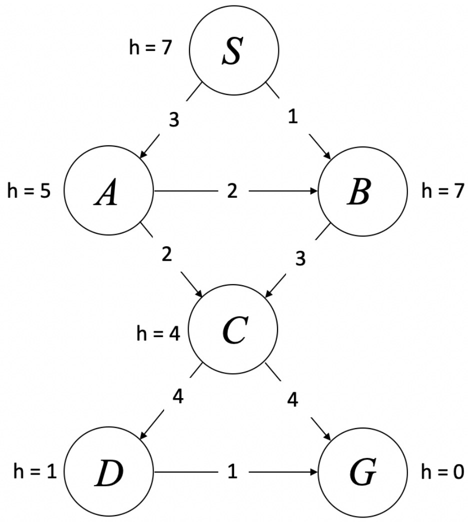
CS 188
Summer 2022

Exam Prep 2A Solutions

Q1. Search Algorithms Potpourri

(a) We will investigate various search algorithms for the following graph. Edges are labeled with their costs, and heuristic values h for states are labeled next to the states. S is the start state, and G is the goal state. In all search algorithms, assume ties are broken in alphabetical order.



(i) Select all boxes that describe the given heuristic values.

■ admissible ■ consistent □ Neither

(ii) Given the above heuristics, what is the order that the states are going to be expanded in, assuming we run A* graph search with the heuristic values provided.

Index	1	2	3	4	5	Not Expanded
S	●	○	○	○	○	○
A	○	●	○	○	○	○
B	○	○	●	○	○	○
C	○	○	○	●	○	○
D	○	○	○	○	○	●
G	○	○	○	○	●	○

(iii) Assuming we run A* graph search with the heuristic values provided, what path is returned?

○ $S \to A \to B \to C \to D \to G$ ○ $S \to A \to C \to G$ ○ $S \to A \to C \to D \to G$
● $S \to B \to C \to G$ ○ $S \to A \to C \to D \to G$ ○ $S \to A \to C \to D \to G$
○ $S \to A \to B \to C \to G$ ○ None of the above

(iv) Given the above heuristics, what is the order that the states are going to be expanded in, assuming we run greedy graph search with the heuristic values provided.

Index	1	2	3	4	5	Not Expanded
S	●	○	○	○	○	○
A	○	●	○	○	○	○
B	○	○	○	○	○	●
C	○	○	●	○	○	○
D	○	○	○	○	○	●
G	○	○	○	●	○	○

(v) What path is returned by greedy graph search?

○ $S \to A \to B \to C \to D \to G$ ● $S \to A \to C \to G$ ○ $S \to A \to C \to D \to G$
○ $S \to A \to C \to D \to G$ ○ $S \to A \to C \to D \to G$ ○ None of the above

(b) Consider a complete graph, K_n, the undirected graph with n vertices where all n vertices are connected (there is an edge between every pair of vertices), resulting in $\binom{n}{2}$ edges. Please select the maximum possible depth of the resulting tree when the following **graph** search algorithms are run(assume any possible start and goal vertices).

	1	$\lceil \frac{n}{2} \rceil$	$n-1$	$\binom{n}{2}$	None of the above
BFS	●	○	○	○	○
DFS	○	○	●	○	○

(c) Given two admissible heuristics h_A and h_B.

(i) Which of the following are guaranteed to also be admissible heuristics?

□ $h_A + h_B$ ■ $\frac{1}{2}(h_A)$ ■ $\frac{1}{2}(h_B)$ ■ $\frac{1}{2}(h_A + h_B)$ □ $h_A * h_B$ ■ $max(h_A, h_B)$
■ $min(h_A, h_B)$

(ii) Consider performing A* **tree** search. Which is generally best to use if we want to expand the fewest number of nodes? Note this was changed from graph to tree search during the exam

○ $h_A + h_B$ ○ $\frac{1}{2}(h_A)$ ○ $\frac{1}{2}(h_B)$ ○ $\frac{1}{2}(h_A + h_B)$ ○ $h_A * h_B$ ● $max(h_A, h_B)$
○ $min(h_A, h_B)$

(d) Consider performing tree search for some search graph. Let $depth(n)$ be the depth of search node n and $cost(n)$ be the total cost from the start state to node n. Let G_d be a goal node with minimum depth, and G_c be a goal node with minimum total cost. Assume edge costs > 0.

(i) For iterative deepening (where we repeatedly run DFS and increase the maximum depth allowed by 1), mark all conditions that are guaranteed to be true for every node n that could be expanded during the search, or mark "None of the above" if none of the conditions are guaranteed.

□ $cost(n) \leq cost(G_c)$
□ $cost(n) \leq cost(G_d)$
■ $depth(n) \leq depth(G_c)$
■ $depth(n) \leq depth(G_d)$
○ None of the above

When running iterative deepening we will explore all nodes of depth k, before we explore any nodes of depth k + 1. As a result will never explore any nodes that have depth greater than G_d because we would stop exploring once we reached G_d. This also means we would never explore any nodes with depth greater than G_c because G_c has to have depth greater than or equal to the minimum depth goal, G_d

(ii) What is necessarily true regarding iterative deepening on any search tree?

■ Complete as opposed to DFS tree search
□ Strictly faster than DFS tree search

- [ ] Strictly faster than BFS tree search
- [x] More memory efficient than BFS tree search
- [ ] A type of stochastic local search
- ○ None of the above